Experimental tests

Here we describe the classification experiment that we carried out to determine the technical requirements to start developing ST4. Our goal is to determine whether a tool that we have selected (detailed below) and other information that we extract from our datasetcould be enough for a dialogue system to classify student answers according to our assessment criteria. Below we provide an overview of what our tests consisted on. Then, we describe in detail the features that were used in the classification tasks. We then describe those classification tasks. After that, we discuss the main results of our tests. This is followed by a more detailed analysis, and a discussion of what these results would imply for the design of the system.

As ST4 is concerned with analyzing students' paraphrase of the main ideas of a text (thesis + arguments), we hypothesized that comparing the answers against the relevant paragraphs could provide the system with the necessary features to evaluate the answers.The tool used for the comparison was a Semantic Textual Similarity (STS) tool developed by the IXA group using a model by Cer et al. (2018), the Universal Sentence Encoder. The model converts sentences into vectors, so the similarity between sentences can be calculated bycomparing their vectors (ibid). Cer et al. demonstrated that their Universal Sentence Encoder can be used for several NLP tasks with good results. In our experiment, we used the STS tool to compare each answer against each paragraph of the text for the exercise that the answer responded to. Answers were also compared against the paragraph function summaries. This way, we were able to obtain scores that reflected how similar a student answer was to each paragraph and their summaries. These scores could then be usedto classify the answers according to our assessment criteria (being on task, being a good paraphrase, being complete and being correct). To do this, we performed four classification tasks, each focusing on labelling answers according to oneof the criteria. For each task, we trained a classifier using the STS scores data, together with some additional information which we discuss below, for part of the answers and teston the remaining answers whether the model learned by the classifier performs adequately.

Features used for classification

The dataset contains answers for ten different texts. These texts have very different structures. In order to test a classifier that could work for any of the texts (texts from our dataset and future texts), we needed the classifier models to use features that were applicable to any of the texts, regardless of structural differences - for example, if the classifiers learn to classify answers for texts where a counterargument is presented before the main argument but not for texts where no counterarguments are mentioned, they will be of little use. For that reason, we created some compound features, turning the varying number of STS values into a constant number of values. The resulting features for each answer ofeach text were as follows:

1. Average STS value of comparing an answer with all the paragraphs containing thethesis

2. STS value of comparing an answer with the thesis summary

3. Average STS value of comparing an answer with all the paragraphs containing arguments
4. Average STS value of comparing an answer with all argument summaries
5. Average STS value of comparing an answer with all paragraphs
6. Maximum STS value of comparing an answer with a paragraph containing the thesis
7. Maximum STS value of comparing an answer with a paragraph containing an argument
8. Maximum STS value of comparing an answer with an argument summary
9. Maximum STS value of comparing an answer with any paragraph
10. Answer word count
11. Unigram overlap between the answer and the whole text
12. Bigram overlap between the answer and the whole text

These last three features, as can be seen, are not related to STS scores, but were addedbecause they seemed potentially very useful and required no more than a simple scriptto be obtained. We believed that the overlap measures would be especially useful for theclassifier evaluating answers with regard to whether the answer was in the student's ownwords or whether it was a poor paraphrase - if the answer has many words in common with the text, then these are not the student's words. We also measured word count thinking that it might be a useful feature for all classifiers, but especially for the one evaluating whether answers are complete - extremely short answers might be less likely to contain the thesis and an argument.

It seems necessary to explain how unigram and bigram overlap measures were calculated, so that the implications of our chosen method can be taken into account. We removed punctuation, as overlap in this respect between the reference texts and the answers would not provide any information useful to our task. To calculate unigram overlap,we also removed stopwords, for the same reason. However, to calculate bigram overlap, we did not remove stopwords, for fear of that resulting in artificial bigrams - technically, removing punctuation could also result in artificial bigrams, but we assumed that the noiseremoved would be greater than the noise introduced. For example, we feared cases where an answer said something as "**volatile gases** are bad", and the reference text said "The situation is **volatile**, as the **gases** in oil..."; removing stopwords could introduce the false bigram "**volatile gases**". Future tests with bigram overlap, and tests on larger n-grams, may reveal whether our precautions were excessive and introduced much useless noise, or whether most noise was as bigrams formed by terms already counted in the unigram overlap together with a function word (e.g. "the volatility", instead of a "true" bigram like "gas volatility"). Whichever the case, noise that simply increases bigram count unnecessarily but does not introduce artificial bigrams may result in no more issue than needing to usea higher threshold to classify answers based on the bigram overlap feature.

Classification tasks

Once all the features were obtained, we performed a series of bi- nary classification tasks. The first classification was to divide answers as on- or off-task. Secondly, we tested how answers could be classified as complete or incomplete (off-task answers were also assigned the "incomplete" target label, together with the answers in the specific "incomplete" category). Another classification test was performed to detect whether answers were in the students' own words or whether they wereloosely copying the text (only answers in the specific "poor paraphrase" category were assigned the negative target label). Finally, answers were to be classified as correct or incorrect (the negative target label was assigned to both the answers in the specific "incorrect" category, as well as the "off-task" answers).

The classification tests were performed using the scikit-learn decision tree classifier tool due to the interpretability of results it offers. The classifier parameters were selected withthe goal of preventing overfitting by controlling the size of the tree. Tests were performedwith ten-fold crossvalidation using 20% test size. We also performed tests dividing the databy document with a simple 80%/20% train/test split, without k-fold crossvalidation, as this allowed us to see individual decision trees, instead of only the general metrics obtained with the crossvalidation tests. The split was random, but coincidentally the two texts of the test set were one with an argument hierarchy and one without, resulting in a balancedsample.

General findings

Table 1 shows the accuracy scores obtained by the decision tree classifier for each of the four binary classification tasks (the scores from the second column are averages for the tencrossvalidation folds, together with the standard deviation (SD)). The tasks are indicated on the left column. Each of the other columns corresponds to a method used to splitthe dataset into a training a testing subset. As explained earlier, we used ten-fold crossvalidation, as well as a simple split of 80% of documents for training, 20% for testing,to see more clearly the impact of individual texts' characteristics on results. The resulting split was balanced with regard to the number of documents with a hierarchy of arguments(as explained earlier, some texts present arguments of equal weight, while others have a hierarchy).

CLASSIFICATION TASK	10-k CV (Avg.)	Split by document
ON/OFF TASK	0.96 (SD:0.01)	0.96
COMPLETENESS	0.85 (SD:0.02)	0.77
OWN WORDS	0.98 (SD:0.01)	0.99
CORRECTNESS	0.85 (SD:0.02)	0.83

Table 1: Accuracy scores for each classification task with two dataset splitting methods

What we can first observe from these accuracy scores is that the features we have used in this initial test might suffice for some of the classification tasks. The easiest classification tasks seem to be determining whether answers are on or off-task and whether

they are good paraphrases: nonetheless, tests with real answers would probably return lower accuracy scores, at least until enough real answers can be gathered to train a new, more sophisticated classifier. We have striven to write diverse, plausible answers, but we cannot anticipate what real answers may truly be like. Students' bad paraphrases may perhaps be less bad than the ones we created with Wordnet, which would complicate the classification. The scores for the other two classification tasks, the ones concerning completeness and correctness, while good, are not as high, even with simple, artificial data. This suggests that the features used for classifying answers with regard to these categories may not be enough; other more adequate models may be required to extract more useful information from students' answers for their classification (e.g. models trained for stance detection might help determine whether students' answers reflect the correct stance from the reference text). Table 1 also shows that classifying answers by completeness is affected by the idiosyncrasy of the reference texts, as gleaned from the lower accuracy scores obtained when the dataset was split by document.

Figure 3 shows the confusion matrices for the classification tasks (with the simple dataset split, using 20% of documents as test set). The matrices show that classification was mostly done well, with some minor exceptions. Firstly, we can observe that, overall, there were very few false negatives - very few good answers were classified as bad. The highest rate of false positives was found in the classification task according to completeness: 5% of answers from the test set. These good results could perhaps be due to the imbalanced categories. Earlier, when discussing the dataset compilation process, we justified the need to work with these imbalanced categories. The scale of this project and the goal of merely getting a broad idea of the STS tool's suitability for ST4 of the task do not warrant an expansion of our experiment to try different methods of artificially balancing the sample distribution. Still, we performed onesuch test to see to what extent the high accuracy in some categories might be due to the imbalance in sample distribution. We took the best-performing category, paraphrasing, where results might be "too good to be true", and up-sampled the minority class (bad paraphrase). Accuracy decreased, but only by 3%, which maintains it above 90%. Thus, it seems that the high accuracy cannot be attributed solely to the imbalanced distribution, but mainly to the quality of the STS model. Nonetheless, once sufficient real answers could be gathered, more realistic class distributions could be achieved. They might still be imbalanced if students' performance is not balanced, and some types might not even appear (e.g. in a study setting, participants might be the people with the most motivationto carry out the task and everyone's performance would be excellent (Pinkwart et al., 2008)). Thus, our limited dataset at least has the advantage of containing a wide varietyof answer types.

If we look at the false positives (bad answers that were classified as good), results are only slightly worse for the on-/off-task and the own words/bad paraphrase classification tasks; for the completeness and correctness classification tasks, results are noticeably worse. Almost 18% of answers from the test set were incorrectly classified as incomplete, and 15% were wrongly classified as correct. While these are not excellent results, especially considering that we used simple, artificial answer data, we believe that, in an educationaltask, false positives are less problematic than false negatives. If a student receives corrective feedback on their good answers, they could get frustrated and unmotivated (Kulatska, 2019). On the other hand, if a student gives a bad answer that is not so clearly bad for the system to label it as such, not giving them feedback might not be so problematic - if their answer was not extremely bad, their need for

feedback may also not be so extreme.

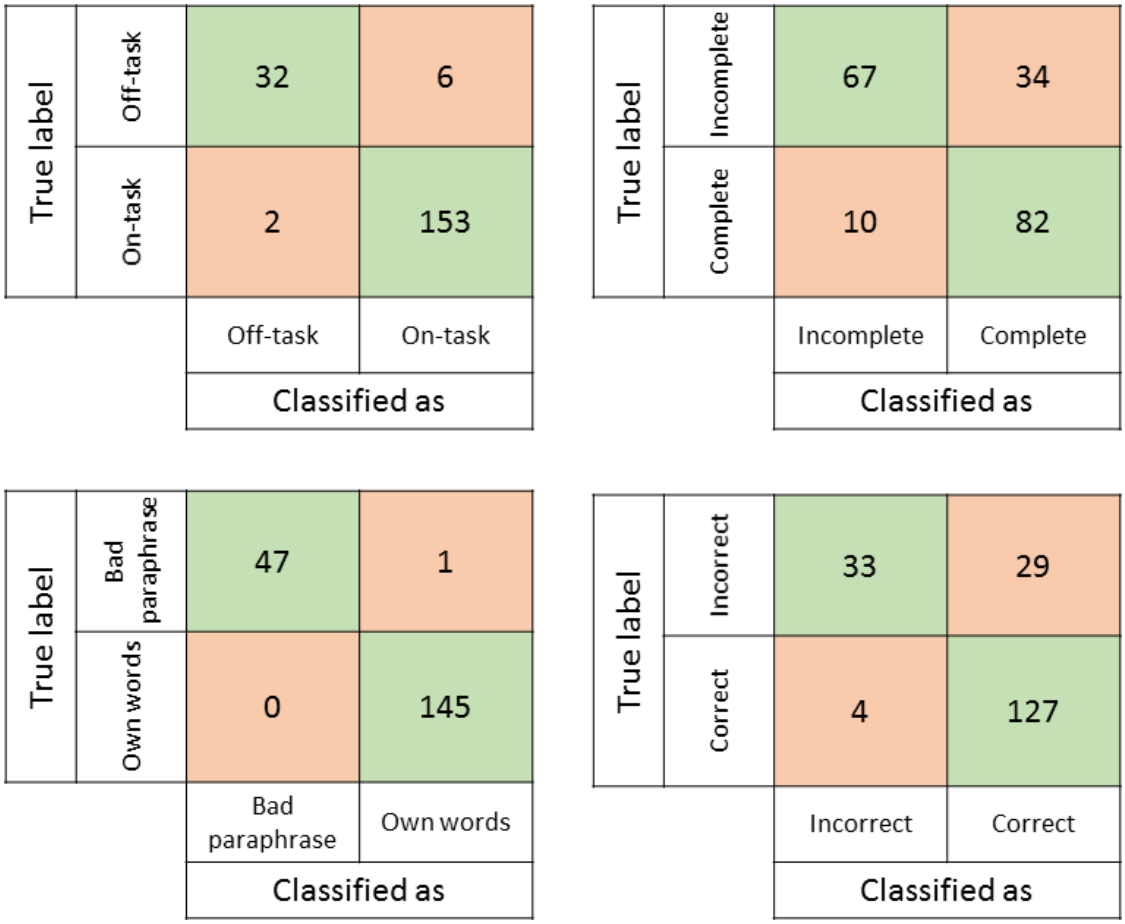

Figure 3: Confusion matrices with dataset split by document

We also analyzed the features that the classifiers found most informative for each of the classification tasks. What first catches our attention is that the function summaries (described in the document detailing the dataset compilation process) turned out to be unnecessary for classification; at least they served to expedite the annotation process. The features most used by the classifiers were 1) the maximum STS score of an answer compared with a paragraph containing the thesisand 2) the word count. The maximum STS score of comparing answers with the whole reference text was also used to classify answers for correctness and whether they were on task. The correctness classifier also used the maximum STS score of answers compared with a paragraph containing an argument. The completeness classifiers used either the maximum of the average STS score of answers compared with a paragraph containingan argument. Bigram overlap was also used in several classifiers, most consistently to classify answers as on or off task. This classification task also used unigram overlap.This contradicts our hypothesis that overlap measures might be most useful for classifyinganswers as good or bad paraphrases - for that classification task,

the most informative feature was the maximum STS score of comparing an answer with paragraphs containing the thesis.

Analysis by assessment criteria

Here we take a closer look at the results for each category. We start with the categories that performed the worst; we believe that they do not warrant an in-depth analysis, as the low performance may be due to the STS model not being ideally suited for them. Nonetheless, we take a general look at these categories and comment on how results couldbe improved in future tests. For the best performing categories, we do provide some more details on how the features were used for classification and provide a graphical example ofhow it works.

Our different splitting methods revealed that completeness is the classification task most sensitive to the reference texts' idiosyncrasy, as seen most evidently on Table 1. Further tests would need to be carried out to arrive at a better way of classifying students'answers with regard to this completeness. We surmise that a model trained to distinguisharguments from unreasoned statements might help - complete answers state a position and back it with at least one argument. However, seeing how sensitive this category is to each document's characteristics, it might still be necessary to combine that model witha similarity model like the one that we have used, or some other, to make sure that the answers are justified with the required elements from the text, and not something else (e.g.a minor argument, or some other argument invented by the student before they are askedto think of their own arguments).

Correctness, while not as problematic as completeness, was also a criterion where classification showed much room for improvement. The confusion matrices show that there could be many false positives in this category - many incorrect answers could be labeled as correct and receive no corrective feedback. We mentioned in section 6.2.3 that it might not be very problematic if some slightly incomplete answers do not receive feedback; however, me must also assume that accuracy would drop when using real student data, and false positives might reach a proportion not appropriate for a pedagogical task like the one we propose. If the students do not receive the feedback they need, they might not be readyto complete the following part of the task - we would be removing the scaffold before the student can work without it. Given that detecting whether an answer is correct or incorrect is, in essence, detecting whether the students' answers have the same stance as the author's text, a model trained for stance detection could improve accuracy.

We now move on to the paraphrasing criteria; the features that we have used seem to be adequate for obtaining good classification performance in this and the on-/off-task criteria. We include Figure 4 as an example of how classification could be done with regard to para- phrasing; we have selected this tree as the most illustrative example of how the classifiers work, because it is the simplest tree. We see that there are two very clear thresholds of similarity with the text's thesis above which the answer can be confidently considered anear copy of the text, and below which the answer can be confidently considered to be inthe student's own words (off-task answers would be the ones lowest below the threshold). Nonetheless, there is a narrow band between those thresholds where classification cannotbe performed so confidently - some good answers would be at this point of being verysimilar to the text, but not so similar that they copy it. Fortunately, this uncertainty was happily resolved in our tests and there were no false negatives (good paraphrases la- belled as bad). There were, however, a small number of false positives (bad paraphrases labelled as good). An analysis of the classification criteria suggests that false positives would mainly be bad paraphrases in the form of short sentences (short instances of the poorpar-ordmore category). The STS model that we have

used thus seems exceedingly sensitive to text length. A look at the longer answers confirms this assumption: very longbad paraphrases were given a similarity score of 1 (the maximum) despite not being exact copies of the text, which is what that maximum similarity score would imply. Nonetheless, the accuracy scores obtained in these initial tests suggest that this or other similar models (perhaps a model trained with sentences and paragraphs of varying length) couldgive adequate results.

We finally look at the on-/off-task classification, which we assumed would be the most

accurate, though the paraphrasing category surpassed our expectations. We have observed a clear threshold of similarity with any paragraph of the reference text above which answers can be confidently classified as on-task; there is also a threshold for unigram overlap that results in zero entropy. However, answers below these thresholds cannot be classified so confidently based on the available features. Unigram overlap can be an informative feature for most cases, but two off-task subcategories present some problems: off-rand (off-task answers containing a domain word or expression) and off-gibb (random answers, which we created by randomly copying song lyrics from different genres). The issues with the off-rand category were expected: these off-task answers were designed to be problematicby including domain words. Therefore, they may have a higher unigram overlap than some on-task answers that use synonyms instead the specific domain words from the text. The issues with the off-gibb category were more unexpected; it seems that removing stop- words is not enough, and perhaps the overlap value should be extracted counting only domain words, which would have to be distinguished from non-domain-specific yet also non-stopword terms. Thus, labelling the text domain (manually or automatically) might be a necessary first step, to then compare only unigrams belonging to that domain (perhaps with a domain-annotated lexical database, such as Wordnet Domains[23]). Difficulties in classifying these and other off-task subcategories also seem to stem from the STS tool'ssensitivity to text length. For example, an answer like "Eh?" is given a similarity as highas 0.538 to a text paragraph, the same as an answer like "The used-clothes industry isnot ethical", which is an on-task answer that is simply too short because it is missing the justification for the thesis. Nonetheless, there were only a few false positives in our tests.

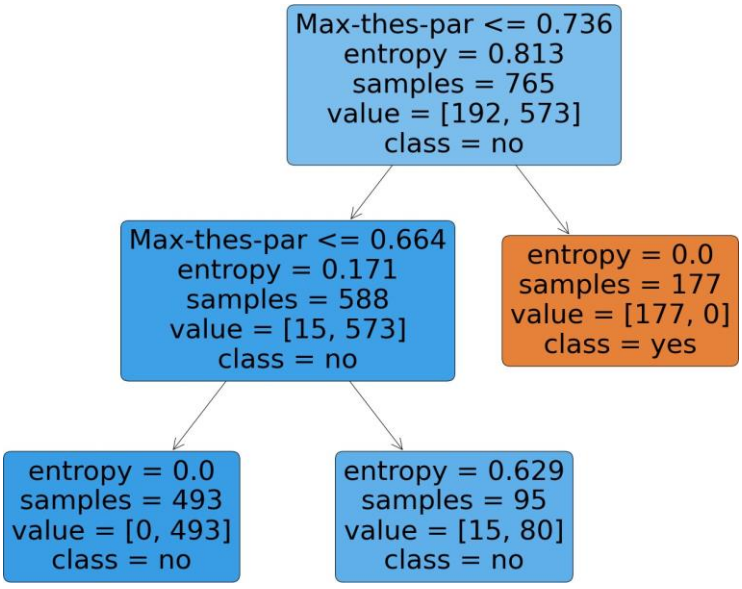

Figure 4: Decision tree example for classification into bad paraphrase/own words. Majority class is own words.

Implications for system design

Firstly, we can conclude that annotating the paragraph function summaries would be unnecessary, even though the STS model used was confirmed to perform unpredictably when the answers were exceedingly shorter than the reference paragraph. The features obtained from the function summaries seemed to provide no usefulinformation for classification. Still, they are useful in the annotation process to provide a quick overview of the reference text.

The classification accuracy scores obtained in the different categories suggests that themodel that we have used could be adequate for determining whether students' answers are on task and whether they are good paraphrases, though it might be necessary to also procure and test a model more adapted to varying text lengths. Perhaps once a first version of the dialogue system was fully developed and sufficient real answers could be gathered, a model could be trained to obtain even better accuracy. The accuracy scores from ourtests also suggest that a different model would be needed to provide sufficiently accurate classification of students' answers with regard to completeness and correctness.

If the dialogue system were to be developed without additional, more adequate models, the hierarchy that we established for the different assessment categories might have to be modified to be more in line with the system's capabilities (the hierarchy was described when detailing the dataset compilation process) . Naturally, our first level in the hierarchy was determining whether the student's answer was an actual attempt at an answer; fortunately, this was done very accurately in our tests, so this level could be maintained in first place. We then set completeness as the second level of the hierarchy: as the previous subtask (ST3) required the student spotting paragraphs that contained the author's thesis and at least an argument, the next logical step might be ensuring that the student used all the elements they had identified. However, given the good-yet-not-ideal accuracy scores obtained for completeness classification, this second level might need to be switched with the third criteria: whether the answer is a good paraphrase. Paraphrasing ideas can be seen as a way of processing and incorporating them (Skidmore, in Mercer et al., 2019); checking whether this processingis taking place could be done before checking whether the student processed all the text elements that they needed to process. Correctness is the aspect where we would expect the fewest issues: a student that has produced a complete answer and paraphrased the text appropriately could be assumed to have done enough processing to not misinterpret thetext. As such, correctness could remain last in the hierarchy. In addition to changing the order in which the different assessment criteria would be checked, the system's potential inaccuracy at some categories could be compensated by taking confidence into account(Jurafsky and Martin, 2019). When an answer did not meet the criteria for confident classification into a category, classifying the answer would run the risk of frustrating the student by giving mistaken corrective feedback (Kulatska, 2019) or of not providing nec essary feedback to a student whose answer could need some improvement. An alternative would be to respond to such answers that cannot be classified confidently with no corrective feedback, but some other non-committal prompt that could guide the students who needed guidance, but which could be ignored by students who gave a perfect answer (e.g. "Not bad :) Did you check if your answer contains the author's thesis and a supporting argument from the text? I forgot to check"). To ensure that

the student is given enough scaffolding, it would also be possible to add a simple multiple-choice task that checkedstudents' grasp of the aspects that the system could not evaluate confidently - though this would imply creating such questions, either manually or automatically.